This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Global Structure-from-Motion by Similarity Averaging

Zhaopeng Cui and Ping Tan
Simon Fraser University
{zhaopeng.cui, pingtan}@sfu.ca

Abstract

Global structure-from-motion (SfM) methods solve all cameras simultaneously from all available relative motions. It has better potential in both reconstruction accuracy and computation efficiency than incremental methods. However, global SfM is challenging, mainly because of two reasons. Firstly, translation averaging is difficult, since an essential matrix only tells the direction of relative translation. Secondly, it is also hard to filter out bad essential matrices due to feature matching failures. We propose to compute a sparse depth image at each camera to solve both problems. Depth images help to upgrade an essential matrix to a similarity transformation, which can determine the scale of relative translation. Thus, camera registration is formulated as a well-posed similarity averaging problem. Depth images also make the filtering of essential matrices simple and effective. In this way, translation averaging can be solved robustly in two convex L_1 optimization problems, which reach the global optimum rapidly. We demonstrate this method in various examples including sequential data, Internet data, and ambiguous data with repetitive scene structures.

1. Introduction

Structure-from-motion (SfM) methods estimate 3D scene structures and camera poses from 2D images. The 'gold standard' algorithm, bundle adjustment (BA)[40], minimizes the reprojection error to achieve a maximum likelihood estimation. SfM methods can be roughly categorized as incremental or global according to their ways to initialize BA. Incremental methods (*e.g*. [38]) initialize cameras one by one. They are typically slow and subject to large drifting errors [9], though impressive results are demonstrated [2] on huge scale Internet image sets. Global methods (*e.g*. [15]) initialize all cameras simultaneously and have better potential in efficiency and accuracy.

Global SfM methods face two major challenges. Firstly, motion averaging, in particular, translation averaging is hard. It is hard to determine global camera positions from local relative translations encoded in epipolar geometry (EG). Many translation estimation methods [15, 4, 3, 29] degenerate at collinear camera motion, because an essential matrix does not tell the scale of translation (*i.e*. the baseline length).

Secondly, global SfM methods are more fragile on noisy data, *e.g*. Internet images, due to poor relative motion estimation caused by feature matching failures. Global methods have to carefully filter out wrong EGs before motion averaging. In comparison, local methods benefit from the RANSAC process to exclude bad feature correspondences when adding additional cameras to the reconstruction. EG filtering is still an open problem despite various consistency filters adopted in [21, 27, 42].

We tackle both problems by constructing a sparse 'depth image' for each camera, which contains depth values at a sparse set of feature points. Depth images upgrade an essential matrix to a similarity transformation, which encodes the relative rotation, translation, and scale between two depth images. The relative scale change encodes the baseline length, so that similarity averaging is much well-posed. In comparison, from essential matrices, translations can only be solved for cameras in a parallel rigid graph [29]. While some methods [34, 26, 12, 42] use depths of scene points to go beyond the parallel rigid graph, they exploit the depth information of a single point at a time and are sensitive to outlier points. Our method based on depth images is a more holistic approach.

Our similarity averaging algorithm includes three steps, *i.e*. rotation averaging, scale averaging, and scale-aware translation averaging. We take the robust method in [8] for rotation averaging. Both our scale and translation averaging are formulated as convex L_1 optimization problems, which converge rapidly to a global optimum.

At the same time, depth images make the filtering of bad EGs easy. During the construction of depth images, we filter pairwise relative motions by the depth consistency of reconstructed feature points, which excludes EGs with large errors. For pathological data with repetitive scene structures (*e.g*. examples published in [33]), our depth images allow straightforward 'missing correspondences' analysis [44] to

discard outlier EGs. These filters make our method robust on challenging data.

In experiments, we demonstrate our algorithm on sequential data, Internet data, and ambiguous data with repetitive structures. Our method consistently outperforms recent global SfM methods [42, 29] and well known incremental methods [38, 43]. In terms of runtime efficiency, our BA initialization is up to 3 or 9 times faster than the methods in [42] and [29] respectively. Our complete system is up to 30 times faster than a parallelized version of Bundler [38].

2. Related Work

Incremental methods [32, 38, 31, 2] add cameras one by one to initialize the final BA. The reconstruction quality heavily depends on the initial pair of cameras and the order of adding other cameras, which might be optimized by the 'next-best-view' algorithms (*e.g*. [13, 18]). Some incremental methods [14, 24] hierarchically merge small reconstructions into larger ones. All incremental methods are subject to significant drifting errors on large image sets, especially when the input EGs are noisy. Frequent intermediate BA reduces drifting but creates computation bottleneck.

Rotation averaging [15] solves all camera orientations simultaneously from input pairwise relative rotations. This problem is complicated due to the nontrivial topology of the rotation manifold [19]. Linear algorithm is presented by ignoring the manifold constraint [26]. Better result is achieved by Lie-algebra representations [16] and further combined with robust L_1 optimization [8]. Rotation averaging is also intensively studied in robotics and control, with an up-to-date survey at [7].

Translation averaging computes camera positions, typically with their orientations fixed beforehand. A key challenge here is that an essential matrix only encodes the direction of translation. Therefore, as rigorously proved in [29], essential matrices only determine camera positions in a parallel rigid graph. As a result, essential matrix based methods [15, 4, 3, 29] are ill-posed at collinear camera motion, which excludes them from many robotics applications where linear motion is common. Cui *et al*. [12] reinforce essential matrices with feature tracks, but require careful feature track filtering.

Trifocal tensor based methods [36, 10, 27, 21] are robust to collinear motion, because the relative scales of translations are encoded in a trifocal tensor. However, they only reconstruct images within a connected camera triplet graph, which is often a much smaller subset of images. As discussed in [12], it is also not easy to balance the number of constraints for different triplets.

Some methods [34, 23, 26, 11, 42, 37] solve scene points and camera poses together. Generally speaking, the L_∞ methods [23, 26] are sensitive to outliers. The discrete-continuous optimization [11] is computationally inefficient.

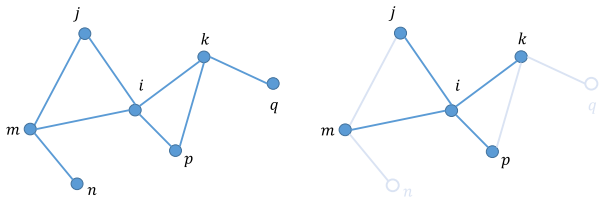


Figure 1. Left: an EG graph where each camera is a vertex and two cameras are connected if the essential matrix between them is known. Right: a stellate graph includes all vertices and edges directly linked to a center vertex i.

The 1DSfM method [42] is more suitable for Internet images which tend to produce $O(n^2)$ essential matrices for n images. The method in [37] adopts the linear constraint in [34] and initializes the solution by stitching pairwise reconstructions. Its initialization bears some similarity to our method. But it achieves poor accuracy as reported in [21, 29]. A key difference is that our method reconstructs 'depth images' to upgrade essential matrices to similarities for well-posed motion averaging. Furthermore, we include EG filters in the construction of depth images, which plays a critical role in the robustness of our algorithm.

Outlier EG filtering is critical for both incremental and global SfM methods. Global SfM methods are more sensitive to this filtering, since incremental methods can benefit from RANSAC based correspondence filtering when adding cameras. Various filters have been designed based on loop consistency check [45], random sampling of EGs [17], 'missing correspondences' [44], and local feature clustering [41]. For densely matched images, Wilson *et al*. [42] design a smart filter based on 1D SfM, which allows straightforward translation averaging. Some challenging pathological data with repetitive scene structures are published in [33]. The 'missing correspondence' analysis, when combined with timestamp [33] or iterative graph optimization [22], successfully reconstructs those challenging data. Our system includes three EG filters applied to depth images, *i.e*. the depth consistency check, the optional local BA, the optional 'missing correspondence' analysis. These filters enable our method to deal with challenging data.

3. Overview

Our input is a set of images with known essential matrices, *e.g*. computed from the five-point algorithm[28, 25]. We aim to solve all camera positions and orientations in a global coordinate system. As shown in Figure 1 (a), these inputs can be represented by an EG graph, where each camera is a vertex and two cameras are connected if the essential matrix between them is known. The essential matrix on an edge (i, j) encodes the relative rotation $\mathbf{R}_{ij}$ and the relative translation direction $\mathbf{t}_{ij}$ of two cameras i and j. We aim to estimate the camera positions $\mathbf{c}_i$ and orientations $\mathbf{R}_i$

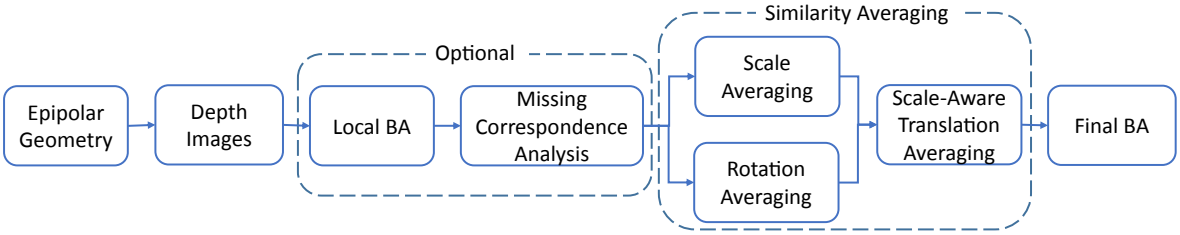


Figure 2. Pipeline of the proposed method. See text for more details.

in a global coordinate system, which are constrained by the following equations,

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^\top, \tag{1}$$

$$\mathbf{t}_{ij} \sim \mathbf{R}_j(\mathbf{c}_i - \mathbf{c}_j). \tag{2}$$

Here, $\sim$ means equal up to a scale.

A key difficulty arises from the fact that $\mathbf{t}_{ij}$ is a unit vector, since the baseline length is not determined by an essential matrix. As rigorously proved in [29], the essential matrices only determine camera positions in a parallel rigid graph, *e.g.* collinear cameras cannot be solved. We seek to bootstrap the global SfM problem by constructing a sparse 'depth image' D_i at each camera i from a stellate graph shown in Figure 1 (b), which includes all vertices and edges directly linked to i. A depth image contains the depth values at sparse features. With these depth images, we can upgrade an essential matrix to a similarity transformation which encodes additional scale changes. The baseline lengths can be easily computed from these scale changes. Therefore, translation averaging becomes a well-posed problem.

Figure 2 provides an overview of our system pipeline. We first construct a depth image for each camera. This step further performs depth consistency check to exclude noisy essential matrices. Following the construction of depth images, there are two optional EG filters. The local BA is applied to images in the local stellate graph to improve pairwise relative motion and also exclude some poor essential matrices. The 'missing correspondence' analysis is applied between image pairs to exclude outlier essential matrices due to repetitive scene structures. In the next, we start to register all cameras in a global coordinate system by a novel similarity averaging. Specifically, we first solve camera orientations by rotation averaging [8]. At the same time, the global scale of each depth image is solved by scale averaging. Once the scales and rotations are fixed, we solve baseline lengths and then camera positions by a scale-aware translation averaging. Once all cameras are fixed, we adopt multiple view triangulation [20] to compute the scene structure, and apply a final BA to optimize cameras and 3D points together.

4. Sparse Depth Image Construction

For each EG edge, we have a local pairwise reconstruction computed by two-view triangulation with the relative pose. The baseline length is set as 1. We build a depth image D_i by stitching the pairwise reconstructions within a stellate graph centered at camera i. Since the stellate graph does not contain any loop, this process is simple and robust. For better computation efficiency, we only consider at most 80 cameras connecting to i. These cameras are selected as those with largest number of feature correspondences.

We put all pairwise reconstructions in the stellate graph under the local coordinate system attached to the camera i. So we only need to solve a scale s_{ij}^i for each image pair (i, j) to stitch these reconstructions together. Here, the upper index ^{i} indicates the stellate graph centered at i. For a feature point in the image i, if it is reconstructed in both image pairs (i, j) and (i, k), its depths relate the scales of both reconstructions as the following,

$$s_{ik}^i / s_{ij}^i = d_{ij}/d_{ik} := d_{jk}^i. \tag{3}$$

Here, d_{ij} and d_{ik} are the depths of that feature point in reconstructions from (i, j) and (i, k) respectively. We apply a median filter to estimate d_{jk}^i from all feature points. Jiang *et al.* [21] compute similar relative scales in a triplet and use them to register cameras in a triplet graph. In comparison, we compute relative scales in a stellate graph and use them to construct depth images. So our method is not limited to triplet graphs.

Taking log of both sides in Equation 3, we have

$$\log(s_{ik}^i) - \log(s_{ij}^i) = \log(d_{jk}^i), \tag{4}$$

which provides a linear equation for the scales of two pairwise reconstructions.

By collecting all such linear equations within the stellate graph, we obtain a large linear equation system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{5}$$

Here, $\mathbf{x}$ and $\mathbf{b}$ are vectors by concatenating $\log(s_{ij}^i)$ and $\log(d_{jk}^i)$ respectively. The matrix $\mathbf{A}$ is sparse. Each row of it contains only two nonzero values 1 and -1. In order to remove the gauge ambiguity, we set the scale of the edge with largest number of matches as a unit, *i.e.* $\log(s_{ij}^i) = 0$, suppose j has the largest number of matches with i.

In order to obtain a robust estimation in presence of outliers, we solve Equation 5 by L_1 optimization as follows,

$$\arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1 . \tag{6}$$

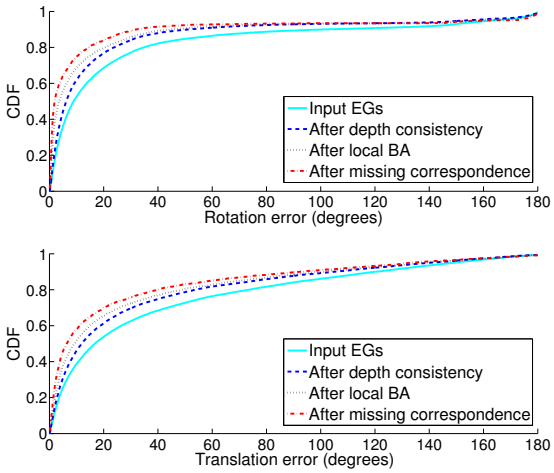


Figure 3. The cumulative distribution function (CDF) of relative motion errors for the *Gendarmenmarkt* data in Section 6.2. The input EGs contain significant errors in both rotation and translation directions. Our depth consistency, local BA, and missing correspondence analysis improves local relative motions for robust global SfM.

This L_1 optimization is convex and achieves the global optimum as studied in [6, 30]. We adopt the efficient package [5] to solve Equation 6.

Depth consistency check Merging the pairwise reconstructions gives multiple depth values for each feature point in camera i. We check their consistency to identify bad EGs. Specifically, we first adopt median filter to compute an optimal depth for each feature point. All depth values deviating more than 5% of the filtered depth are considered as outliers. We remove an image pair (i, j) as a bad EG if it produces less than 5 inliers, because 5 points determine an essential matrix.

To examine the effectiveness of this depth consistency check, we visualize the errors in local relative motion for the data *Gendarmenmarkt* (see Section 6.2) in Figure 3. We take the result from an incremental method [38] as reference 'ground truth' and plot the cumulative distribution function (CDF) of errors in relative rotations and translation directions. As shown in Figure 3, depth consistency check excludes bad EGs and hence improves the robustness of the following motion averaging.

Optional local BA We can optionally apply a local BA to all the cameras and edges in the stellate graph. Unlike the intermediate BA in incremental SfM methods, this local BA is efficient and can be easily parallelized. In our experiment, it takes less than one second to optimize a depth image since the size of a stellate graph is small. This local BA refines the relative motion $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ between camera pairs. Furthermore, after local BA, we discard feature points with large re-projection error, *e.g.* 16 pixels in all our experiments. We then remove an image pair if the number of reconstructed features is less than 5. As shown in Figure 3, the local BA further improves local relative motions.

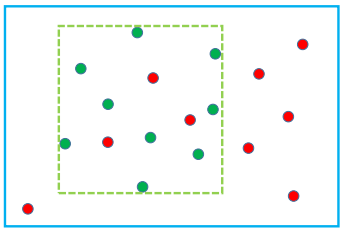


Figure 4. Missing correspondence analysis. The blue frame indicates the field-of-view(FOV) of the camera. Green and red dots are matched and missing features. See text for more details.

Optional missing correspondence analysis 'Missing correspondence' analysis [44] is proved to be an effective way to filter EGs for the challenging data with large repetitive scene structures [33, 22], and it is straightforward with depth images. For an image pair (i, j), we project all the 3D points in D_i to the image plane of j. As illustrated in Figure 4, we visualize these projected points as green and red dots within the image frame. Note we ignore the visibility test and assume all points within the field-of-view (FOV) of j are visible. The green dots indicate points that are matched with features in j. The 'missing correspondences' are points without matches in j, *i.e.* the red dots in Figure 4. The ratio of these red and green dots are analyzed in [44, 22] to discard outlier EGs due to repetitive scene structures.

We take the bounding box (the green dashed line box in Figure 4) of the matched features. We consider missing correspondences, *i.e.* red dots, within this bounding box are due to the imperfect repeatability [35] of feature matching. Thus, we only consider red dots outside of the bounding box as true missing correspondences. We threshold the ratio of these points among all red dots to decide if an edge (i, j) is an outlier. Specifically, we evaluate $M_j^i = n_1/n_2$. Here, n_1 is the number of red dots outside of the bounding box, and n_2 is the total number of red dots. If $M_j^i > \epsilon$, we consider (i, j) as outlier and remove it. We set ϵ to 0.2 for Internet data and 0.1 otherwise. As shown in Figure 3, missing correspondence analysis removes outlier EGs with large errors.

5. Similarity Averaging

Once the depth images are constructed, we compute a similarity transformation for each edge (i, j) of the EG graph. In principal, this similarity can be computed from 3D-3D correspondences between depth images. In practice, we find that the local reconstructions are not precise enough to allow accurate 3D-3D registration. So we keep the relative rotation and translation $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ from essential matrices or after the local BA if it is turned on, and estimate a relative scale S_{ij} simply by

$$S_{ij} = s_{ji}^j / s_{ij}^i, \tag{7}$$

where s_{ij}^i and s_{ji}^j are the scales for (i, j) computed by solving Equation 6 during the construction of D_i and D_j.

Putting the relative rotation, translation, and scaling together, we obtain a local relative similarity transformation $(\mathbf{R}_{ij}, \mathbf{t}_{ij}, S_{ij})$ for any edge (i, j) in the EG graph. We then solve the similarity averaging problem to determine all camera poses. Intuitively, this amounts to stitch all depth images together to form a global 3D reconstruction. Our similarity averaging includes three steps: rotation averaging, scale averaging, and scale-aware translation averaging.

We adopt the robust method proposed in [8] for rotation averaging.

5.1. Robust Scale Averaging

We compute a scaling factor s_i for each depth image D_i to register them together. According to the known pairwise relative scale, we have

$$s_i/s_j = S_{ij}. \tag{8}$$

Taking $\log$ of both sides in Equation 8 gives us

$$\log(s_i) - \log(s_j) = \log(S_{ij}). \tag{9}$$

Collecting this equation from all edges in the EG graph, we stack them into a linear equation system

$$\mathbf{A}_s\mathbf{x}_s = \mathbf{b}_s, \tag{10}$$

where $\mathbf{x}_s$ and $\mathbf{b}_s$ are vectors by concatenating $\log(s_i)$ and $\log(S_{ij})$ respectively. $\mathbf{A}_s$ is a sparse matrix similar to $\mathbf{A}$ in Equation 5. To remove gauge ambiguity, we set the scaling factor of the first image as unit, *i.e.* $\log(s_1) = 0$. We then solve Equation 10 by the following convex L_1 optimization,

$$\arg\min_{\mathbf{x}_s} \|\mathbf{A}_s\mathbf{x}_s - \mathbf{b}_s\|_1 . \tag{11}$$

5.2. Robust Scale-Aware Translation Averaging

Once the global scaling factor of each depth image is determined, we compute the baseline length as,

$$b_{ij} = \frac{1}{2}(s_i s_{ij}^i + s_j s_{ij}^j). \tag{12}$$

With global camera orientations computed from [8], we obtain a linear equation of camera positions as,

$$\mathbf{R}_j(\mathbf{c}_i - \mathbf{c}_j) = b_{ij}\mathbf{t}_{ij}. \tag{13}$$

Collecting this equation from all edges in the EG graph, we can form a linear system,

$$\mathbf{A}_c\mathbf{x}_c = \mathbf{b}_c, \tag{14}$$

where $\mathbf{x}_c$ and $\mathbf{b}_c$ are vectors formed by concatenating $\mathbf{c}_i$ and $b_{ij}\mathbf{t}_{ij}$ respectively. $\mathbf{A}_c$ is a sparse matrix, where each three consecutive rows are all zeros except two rotation matrices $\mathbf{R}_j$ and $-\mathbf{R}_j$. We remove gauge ambiguity by fixing the first camera at original, *i.e.* $\mathbf{c}_1 = \mathbf{0}$. All camera positions are then solved by the following convex L_1 problem,

$$\arg\min_{\mathbf{x}_c} \|\mathbf{A}_c\mathbf{x}_c - \mathbf{b}_c\|_1 . \tag{15}$$

Note the methods in [27, 29] also solve Equation 13 to estimate camera positions. They solve the baseline length b_{ij} simultaneously with camera positions. Their formulation is complicated by the quadratic constraint $b_{ij}^2 = ||\mathbf{c}_i - \mathbf{c}_j||^2$. Typically, this constraint is ignored to simplify optimization. In comparison, we solve b_{ij} beforehand to make the translation averaging simple.

6. Experiment

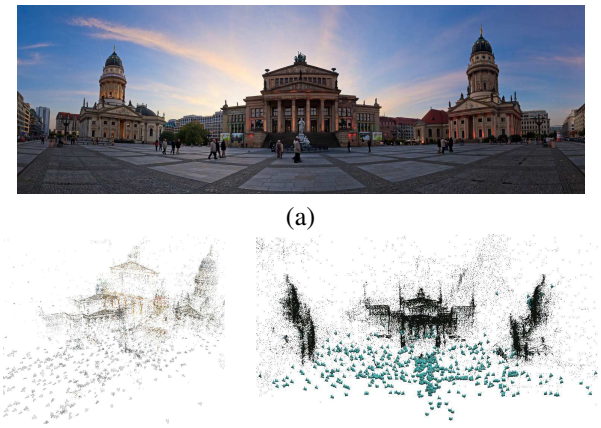
We evaluate our method on various datasets including sequential data, Internet data and challenging pathological data with large duplicate structures. All experiments are run on a machine with two 2.3Hz Intel Xeon E5-2650 processors with 16 threads enabled in total. The depth image generation is parallelized using OpenMP. We use the Ceres solver [1] for the final BA.

6.1. Evaluation on Sequential Data

Global SfM methods solve all camera poses simultaneously from all available relative motions. Thus, they are more robust to drifting problems compared with incremental methods. We demonstrate this on four small to large scale sequential data. The *Herz-Jesu-P25* data from [39] and the *Building* data from [45] has 25 and 128 images respectively. The *Pittsburgh* data consists of 388 Google street view images; The *Campus* data is captured by a smartphone with 1040 images. We compare our method with a typical incremental method, VisualSFM [43], and two recent global methods [29, 42] on these four datasets. The results are shown in Figure 5 and Figure 6.

Figure 5 shows that VisualSFM [43] has good performance when the EGs are good or the sequence is short, while it suffers from severe drifting problem for long sequences with noisy EGs. There are some images with glass walls in the *Pittsburgh* data and many images with trees in the *Campus* data. The quality of pairwise relative motions for these two datasets are poor. So sequential methods like VisualSFM [43] suffer from large drifting errors on these two datasets. Note the loop closure constraints have been provided to VisualSFM in this experiment. But the large drifting error makes the loop closure unsuccessful. The method in [29] generates distortions for the *Pittsburgh* and *Campus* data, because the near collinear motion in these examples makes its essential matrix based translation estimation degenerate.

Figure 6. A failure case of 1DSfM [42] on the *Herz-Jesu-P25* data.



(a) (b) (c)

Figure 8. Results on the challenging *Gendarmenmarkt* data. The image in (a) shows the bilaterally symmetric architecture layout. The results cited from 1DSfM[42] and our method are shown at (b) and (c). Our method succeeds on this data thanks for the EG filtering in local BA and 'missing correspondence' analysis.

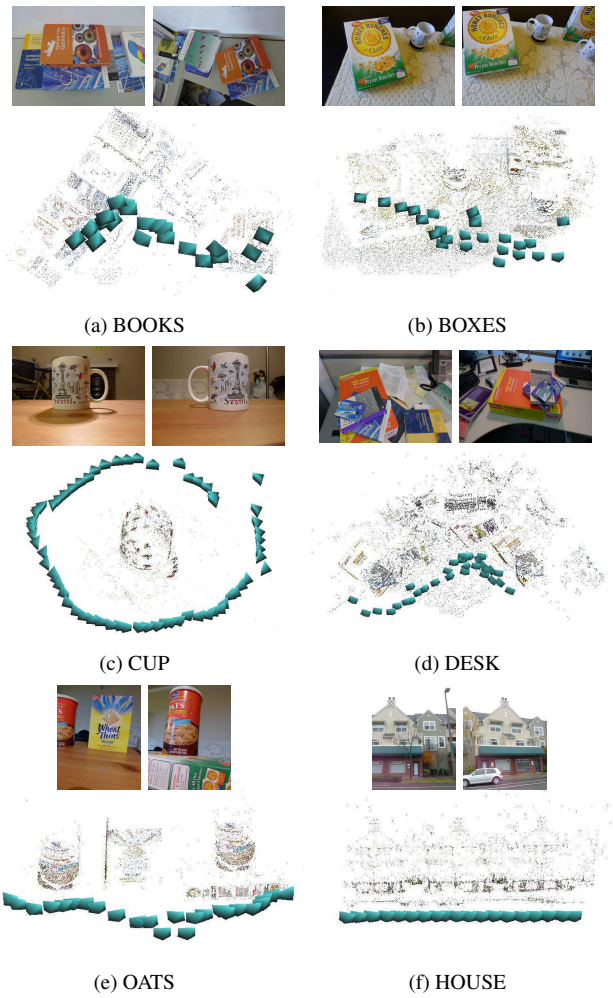
The 1DSfM [42] failed on most of these examples, because it is designed for Internet images that tend to have $O(n^2)$ essential matrices for n input images. Thus, sequential images with $O(n)$ essential matrices is not suitable for this method [1]. Its result for *Herz-Jesu-P25* is shown in Figure 6. Distortions in camera motion can be observed by comparing it with the results in Figure 5.

In comparison, our method performs well for all these examples. It has no visible drifting errors and deals with collinear motions.

6.2. Evaluation on Internet Data

Our method adopts robust optimization in every step and has better performances compared with previous global SfM methods. We demonstrate this on the medium- to large-scale Internet datasets recently published in [42]. We turn on missing correspondence analysis only for the data *Gendarmenmarkt*. The results on the *Piccadilly* and *Trafalgar* data are shown in Figure 7. Table 1 provides quantitative comparison with several global methods [21, 42, 29, 12]. We use the results of an optimized incremental SfM system based on Bundler [38] as the reference 'ground-truth' and compute the mean and median camera position

[1]This is according to our discussion with the authors of [42]



(a) BOOKS (b) BOXES (c) CUP (d) DESK (e) OATS (f) HOUSE

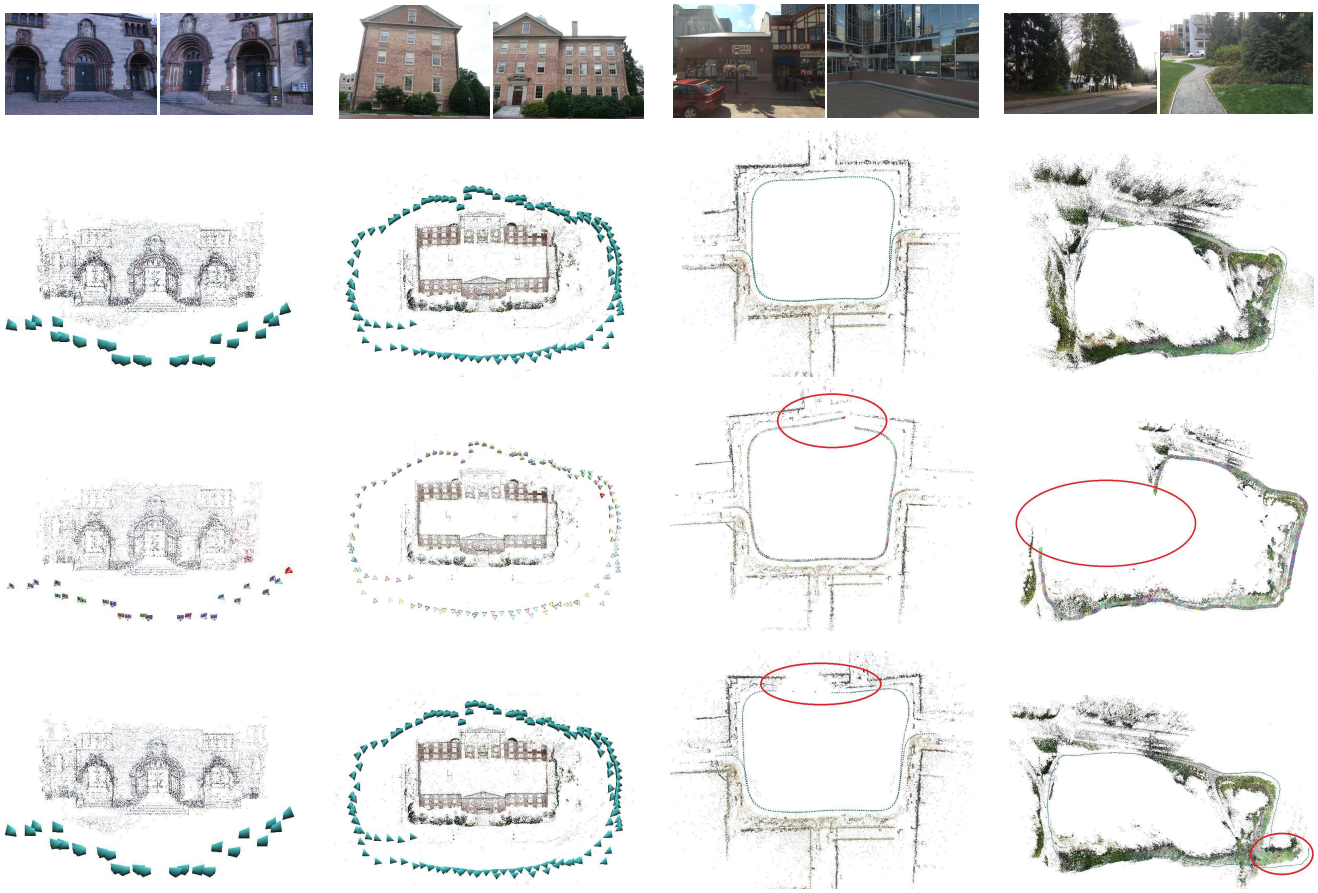
Figure 9. Our results on challenging pathological data with large repetitive structures.

errors for evaluation. The number of reconstructed cameras are also listed for comparison. The results of all other methods are cited from corresponding papers.

From Table 1, we can see that our method with local BA generally has the best accuracy (before the final BA) and reconstructs the largest number of cameras. After final BA, all these methods achieve similar accuracy. Our method without local BA also produces good results, which demonstrates the robustness of our similarity averaging.

We also evaluate the runtime efficiency of these methods in Table 2. Running times of all individual steps are provided for our method. Even with local BA, our method is several times faster than other global SfM methods. For example, our method is about 9 times faster than [29] in initializing BA (*i.e.*, $T_\Sigma - T_{BA}$) on the *Vienna Cathedral* data. It is about 30 times faster than the parallelized version of Bundler [38] on the *Piccadilly* data.

The *Gendarmenmarkt* data is reported as a failure case in [42], due to its repetitive scene structures. None of the previous global SfM methods can reconstruct it successfully.



(a) *Herz-Jesu-P25* (b) *Building* (c) *Pittsburgh* (d) *Campus*

Figure 5. Evaluation on sequential data. From top to bottom, each row shows sample input images, 3D reconstructions generated by our method, VisualSFM [43], and the least unsquared deviations (LUD) method [29] respectively.

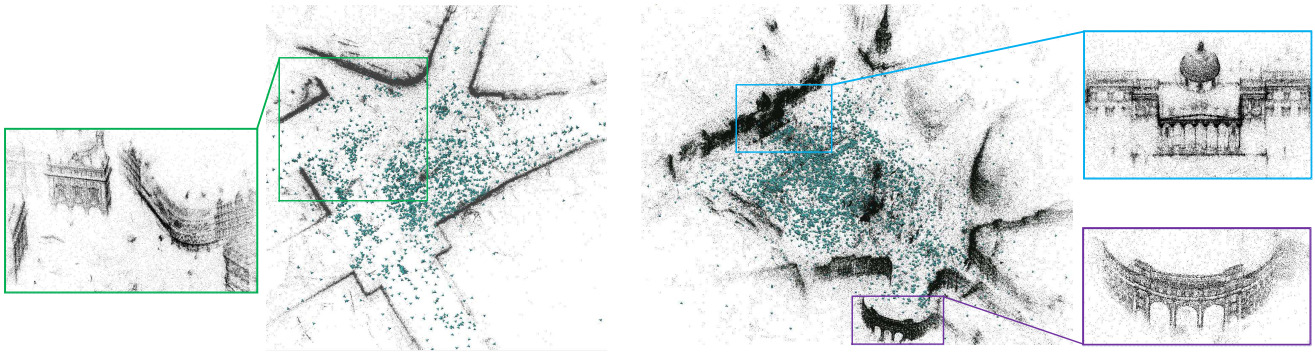


Figure 7. Sample results on the Internet data published in [42]. The left and right are the results on the *Piccadilly* and *Trafalgar* data with 2276 and 4945 images reconstructed respectively.

With local BA and missing correspondence analysis turned on, our method successfully reconstructs this example. We compare our result with that cited from [42] in Figure 8.

6.3. Evaluation on Ambiguous Data

We demonstrate the robustness of our method on some challenging pathological data with large repetitive structures published by [33]. As shown in Figure 9, our method successfully reconstructs all six examples when missing correspondence analysis is turned on.

6.4. Discussions

We further test our method on the very challenging *Quad* data which consists of 6514 images. Our method generates a distorted result for this example. But it produces a reasonable result as shown in Figure 10 when camera orientations

Data		Jiang [21]		1DSfM [42]		LUD [29]			Cui [12]		Ours without LBA			Ours with LBA				
Name	N_i	N_c	$\tilde{x}$	N_c	$\tilde{x}$	N_c	$\tilde{x}$	$\bar{x}$	N_c	$\tilde{x}$	N_c	$\tilde{x}$	$\bar{x}$	N_c	$\tilde{x}$	$\bar{x}$	$\tilde{x}^*$	$\bar{x}^*$
Alamo	613	478	0.6	529	1.1	547	**0.4**	**2.0**	500	0.6	**577**	0.5	**2.0**	574	0.5	**2.0**	0.5	3.1
Ellis Island	242	205	3.2	214	3.7	-	-	-	211	3.1	220	3.3	6.7	**223**	**2.5**	**5.5**	0.7	4.2
Metropolis	384	92	1.7	291	9.9	288	**1.6**	**4.0**	-	-	**317**	3.8	10.8	**317**	2.7	10.6	3.1	16.6
Montreal N.D.	467	333	0.5	427	2.5	435	0.5	1.0	426	0.8	450	0.5	0.8	**452**	**0.4**	**0.7**	0.3	1.1
Notre Dame	552	518	0.4	507	10.0	536	0.3	0.8	539	0.3	547	0.3	**0.6**	**549**	**0.2**	**0.6**	0.2	1.0
NYC Library	369	245	1.0	295	2.5	320	2.0	6.0	288	1.4	337	1.1	2.6	**338**	**0.8**	**1.9**	0.3	1.6
Piazza del Popolo	350	248	**1.1**	308	3.1	305	1.5	5.0	294	2.6	336	3.1	3.9	**340**	2.0	**2.7**	1.6	2.5
Piccadilly	2468	423	2.8	1956	4.1	-	-	-	-	-	2271	1.8	3.1	**2276**	**1.3**	**2.5**	0.4	2.2
Roman Forum	1122	696	13.6	989	6.1	-	-	-	-	-	1069	3.9	10.7	**1077**	**2.9**	**9.4**	2.5	10.1
Tower of London	499	386	5.0	414	11.0	425	4.7	20.0	393	4.4	463	4.0	13.5	**465**	**1.9**	**11.2**	1.0	12.5
Union Square	680	119	**4.3**	**710**	5.6	-	-	-	-	-	570	6.9	15.5	570	5.5	**12.7**	3.2	11.7
Vienna Cathedral	897	478	6.9	770	6.6	750	5.4	10.0	578	3.5	830	2.8	7.2	**842**	**2.7**	**5.9**	1.7	4.9
Yorkminster	450	223	2.6	401	3.4	404	2.7	**5.0**	341	3.7	**419**	2.6	7.2	417	**2.3**	5.7	0.6	14.2
Trafalgar	5288	1481	**3.8**	4591	-	-	-	-	-	-	4881	7.4	11.3	**4945**	5.4	**8.9**	3.6	8.6
Gendarmenmarkt	733	260	23.2	-	-	-	-	-	-	-	-	-	-	**609**	**5.4**	**27.7**	4.2	27.3

Table 1. Comparison on Internet data. $\tilde{x}$ and $\bar{x}$ denote the median and mean position errors in meters for different methods by taking the result of [38] as a reference. $\tilde{x}^*$ and $\bar{x}^*$ denote the median and mean position errors for our method after the final bundle adjustment. N_i is the number of cameras in the largest connected component of our input EG graph, and N_c is the number of reconstructed cameras. For [21], the model with the largest number of cameras is considered. The bold font highlights the best result in each row.

Data		Ours								1DSfM [42]		LUD [29]		Jiang [21]		[38]
Name	N_i	T_D	T_{LBA}	T_{MC}	T_s	T_R	T_c	T_{BA}	T_Σ	T_{BA}	T_Σ	T_{BA}	T_Σ	T_{BA}	T_Σ	T_Σ
Alamo	613	15	52	0	2	5	9	481	578	752	910	133	750	162	191	1654
Ellis Island	242	2	32	0	1	1	2	169	208	139	171	-	-	616	621	1191
Metropolis	384	3	24	0	1	3	3	25	60	201	244	38	142	112	121	1315
Montreal N.D.	467	6	51	0	1	2	4	613	684	1135	1249	167	553	1593	1619	2710
Notre Dame	552	12	49	0	2	3	7	461	552	1445	1599	126	1047	1286	1351	6154
NYC Library	369	2	32	0	1	2	3	171	213	392	468	54	200	464	471	3807
Piazza del Popolo	350	4	36	0	1	2	2	147	194	191	249	31	162	632	643	1287
Piccadilly	2468	16	191	0	11	67	110	1053	1480	2425	3483	-	-	3755	3817	44369
Roman Forum	1122	8	91	0	3	8	29	339	491	1245	1457	-	-	1080	1124	4533
Tower of London	499	3	43	0	1	3	5	503	563	606	648	86	228	2432	2456	1900
Union Square	680	1	34	0	1	3	5	47	92	340	452	-	-	12	14	1244
Vienna Cathedral	897	13	89	0	2	8	13	440	582	2837	3139	208	1467	1145	1187	10276
Yorkminster	450	3	38	0	1	2	4	611	663	777	899	148	297	391	401	3225
Trafalgar	5288	62	379	0	54	181	529	1581	2901	-	12240	-	-	2618	2881	29160
Gendarmenmarkt	733	4	67	1	1	2	6	131	214	-	-	-	-	193	198	-

Table 2. Running times in seconds for Internet data. We report time spent on each step of our method, including depth image reconstruction (T_D), local bundle adjustment (T_{LBA}), missing correspondence analysis (T_{MC}), rotation averaging (T_R), scale averaging (T_s), scale-aware translation averaging (T_c), final bundle adjustment (T_{BA}), and total running times (T_Σ). We cite the final bundle adjustment time and total running time from [42], [29], and [38] for a comparison. Ceres [1] is adopted to solve the final BA for all methods except [29].

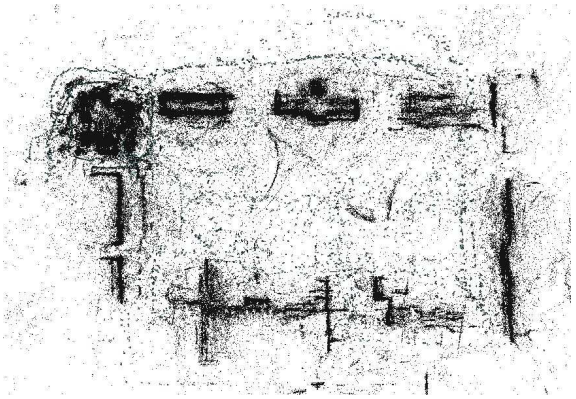


Figure 10. Our result on the *Quad* data. The camera orientations are computed from Bundler[38].

from Bundler [38] is used (instead of those from rotation averaging [8]).

7. Conclusion

This paper presents a novel framework for global SfM. By constructing a sparse depth image at each camera, essential matrices are upgraded to similarity transformations. The camera pose estimation from similarity transformations are well-posed, while essential matrices only determine cameras in a parallel rigid graph [29]. The construction of depth images also makes EG filtering simple and effective. This novel global SfM framework generates superior results in both reconstruction accuracy and computation efficiency.

Acknowledgments We thank Kyle Wilson and Noah Snavely for helpful disccussions. This work is supported by the NSERC Discovery grant 611664 and Discovery Acceleration Supplements 611663.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 5, 8

[2] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *Proc. ICCV*, 2009. 1, 2

[3] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *Proc. 3DPVT*, 2012. 1, 2

[4] M. Brand, M. Antone, and S. Teller. Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *Proc. ECCV*, 2004. 1, 2

[5] E. Candes and J. Romberg. l_1-magic: Recovery of sparse signals via convex programming. http://users.ece.gatech.edu/~justin/l1magic. 4

[6] E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005. 4

[7] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3d slam: a survey on rotation estimation and its use in pose graph optimization. 2

[8] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proc. ICCV*, pages 521–528, 2013. 1, 2, 3, 5, 8

[9] K. Cornelis, F. Verbiest, and L. Van Gool. Drift detection and removal for sequential structure from motion algorithms. *IEEE Trans. PAMI*, 26(10):1249–1259, 2004. 1

[10] J. Courchay, A. S. Dalalyan, R. Keriven, and P. Sturm. Exploiting loops in the graph of trifocal tensors for calibrating a network of cameras. In *Proc. ECCV*, pages 85–99, 2010. 2

[11] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, pages 3001–3008, 2011. 2

[12] Z. Cui, N. Jiang, C. Tang, and P. Tan. Linear global translation estimation with feature tracks. In *Proc. BMVC*, 2015. 1, 2, 6, 8

[13] E. Dunn and J.-M. Frahm. Next best view planning for active model improvement. In *Proc. BMVC*, pages 1–11, 2009. 2

[14] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. ECCV*, pages 311–326, 1998. 2

[15] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proc. CVPR*, pages 218–225, 2001. 1, 2

[16] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. CVPR*, 2004. 2

[17] V. M. Govindu. Robustness in motion averaging. In *Proc. ACCV*, 2006. 2

[18] S. Haner and A. Heyden. Covariance propagation and next best view planning for 3d reconstruction. In *Proc. ECCV*, pages 545–556, 2012. 2

[19] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, pages 1–39, 2013. 2

[20] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 3

[21] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In *Proc. ICCV*, 2013. 1, 2, 3, 6, 8

[22] N. Jiang, P. Tan, and L. F. Cheong. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In *Proc. CVPR*, pages 1458–1465, 2012. 2, 4

[23] F. Kahl and R. Hartley. Multiple view geometry under the l_∞-norm. *IEEE Trans. PAMI*, 30:1603–1617, 2007. 2

[24] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. PAMI*, 27(3):418–433, 2005. 2

[25] H. Li and R. Hartley. Five-point motion estimation made easy. In *Proc. ICPR*, pages 630–633, 2006. 2

[26] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. CVPR*, pages 1–8, 2007. 1, 2

[27] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proc. ICCV*, 2013. 1, 2, 5

[28] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. PAMI*, 26:756–777, 2004. 2

[29] O. Ozyesil and A. Singer. Robust camera location estimation by convex programming. In *Proc. CVPR*, 2015. 1, 2, 3, 5, 6, 7, 8

[30] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013. 4

[31] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3):143–167, 2008. 2

[32] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59:207–232, 2004. 2

[33] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly. Structure from motion for scenes with large duplicate structures. In *Proc. CVPR*, 2011. 1, 2, 4, 7

[34] C. Rother. *Multi-View Reconstruction and Camera Recovery using a Real or Virtual Reference Plane; PHD THESIS*. PhD thesis, January. 1, 2

[35] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151–172, 2000. 4

[36] K. Sim and R. Hartley. Recovering camera motion using l∞ minimization. In *Proc. CVPR*, pages 1230–1237, 2006. 2

[37] S. N. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In *ECCV Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010. 2

[38] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. on Graph.*, 25:835–846, 2006. 1, 2, 4, 6, 8

[39] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proc. CVPR*, 2008. 5

[40] B. Triggs, P. Mclauchlan, R. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. *Lecture Notes in Computer Science*, pages 298–375, 2000. 1

[41] K. Wilson and N. Snavely. Network principles for sfm: Disambiguating repeated structures with local context. In *Proc. ICCV*, pages 513–520. IEEE, 2013. 2

[42] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proc. ECCV (3)*, pages 61–75, 2014. 1, 2, 5, 6, 7, 8

[43] C. Wu. Visualsfm: A visual structure from motion system. 2, 5, 7

[44] C. Zach, A. Irschara, and H. Bischof. What can missing correspondences tell us about 3d structure and motion? In *Proc. CVPR*, 2008. 1, 2, 4

[45] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *Proc. CVPR*, 2010. 2, 5